

## IN THE CLAIMS

1. (Previously Presented) A computer-implemented method, comprising:  
requesting to access a hardware device of a computer system during operating system (OS) runtime of an operating system executing on the computer system;  
accessing the hardware device using an OS native driver for the hardware device if the OS native driver is available to the operating system;  
finding a pre-boot driver for the hardware device by the computer system if the OS native driver is not available to the operating system; and  
accessing the hardware device during the OS runtime using the pre-boot driver if the OS native driver is not available to the operating system.
2. (Original) The method of claim 1, further comprising advertising the pre-boot driver in a data structure by firmware of the computer system, the pre-boot driver to be available to the operating system during OS runtime.
3. (Previously Presented) The method of claim 2 wherein finding the pre-boot driver comprises searching for the pre-boot driver in the data structure.
4. (Original) The method of claim 2 wherein the data structure is compatible with firmware that operates in accordance with an Extensible Firmware Interface (EFI) framework standard.
5. (Original) The method of claim 1 wherein accessing the hardware device comprises executing a pre-boot driver image of the pre-boot driver in a pre-boot driver interpreter operating in the operating system environment.
6. (Original) The method of claim 5 wherein the pre-boot driver image includes interpreted code to allow management of the pre-boot driver image one op-code at a time.
7. (Currently Amended) The method of claim 5 wherein the pre-boot driver interpreter is operated in a pre-boot emulator to emulate ~~the~~a pre-boot environment of the computer system.

8. (Original) The method of claim 5, further comprising stopping execution of the pre-boot driver image if the pre-boot driver image attempts to violate a policy condition of the operating system.
9. (Previously Presented) The method of claim 1 wherein the pre-boot driver comprises an EFI Byte Code (EBC) image for the hardware device.
10. (Canceled)
11. (Original) The method of claim 1, further comprising loading a pre-boot driver image of the pre-boot driver into a memory device of the computer system during a pre-boot phase of the computer system.
12. (Previously Presented) An article of manufacture comprising:
  - a machine-readable medium including a plurality of instructions which when executed by a processor perform operations comprising:
    - discovering a pre-boot driver of a hardware device during a pre-boot phase of a computer system;
    - loading a pre-boot driver image corresponding to the pre-boot driver into a memory device of a computer system;
    - advertising the pre-boot driver in a data structure of the computer system, the data structure available to an operating system of the computer system during OS runtime;
    - accessing the hardware device using an OS native driver for the hardware device if the OS native driver is available to the operating system; and
    - accessing the hardware device during the OS runtime using the pre-boot driver if the OS native driver for the hardware device is not available to the operating system.
13. (Previously Presented) The article of manufacture of claim 12 wherein discovering the pre-boot driver comprises initializing the hardware device during the pre-boot phase having stored the pre-boot driver.
14. (Original) The article of manufacture of claim 12 wherein the hardware device

comprises a non-volatile storage device to store the pre-boot driver.

15. (Original) The article of manufacture of claim 12 wherein the pre-boot driver image comprises an interpreted pre-boot driver image.

16. (Previously Presented) The article of manufacture of claim 12 wherein the plurality of instructions is to operate in accordance with an Extensible Firmware Interface (EFI) framework standard.

17. (Original) The article of manufacture of claim 12 wherein the data structure is compatible with an Extensible Firmware Interface (EFI) framework standard.

18. (Previously Presented) An article of manufacture comprising:  
a machine-readable medium including a plurality of instructions which when executed by a processor perform operations comprising:  
receiving a request from an application executing on a computer system to access a hardware device of the computer system;  
accessing the hardware device using an OS native driver for the hardware device if the OS native driver is available to the operating system;  
finding a pre-boot driver for the hardware device advertised in a data structure of the computer system if the OS native driver is not available to the operating system; and  
executing a pre-boot driver image corresponding to the pre-boot driver via a pre-boot driver interpreter during operating system runtime if the OS native driver is not available to the operating system.

19. (Original) The article of manufacture of claim 18 wherein receiving the request comprises receiving an Application Program Interface (API) call from the application.

20. (Previously Presented) The article of manufacture of claim 18 wherein the pre-boot driver image comprises an interpreted pre-boot driver image to enable access to the hardware device.

21. (Previously Presented) The article of manufacture of claim 18 wherein the data

structure is compatible with an Extensible Firmware Interface (EFI) framework standard.

22. (Previously Presented) The article of manufacture of claim 18 wherein the pre-boot interpreter is to enable managed execution of the pre-boot driver image, wherein the pre-boot driver image includes interpreted code.

23. (Original) The article of manufacture of claim 22 wherein the execution of the plurality of instructions further perform operations comprising stopping execution of the pre-boot driver image if the pre-boot driver image attempts to violate a policy condition of the computer system.

24. (Original) The article of manufacture of claim 18 wherein execution of the plurality of instructions further perform operations comprising executing a native driver for the hardware device if the native driver for the hardware device is available.

25. (Previously Presented) A computer system, comprising:  
a processor;  
an expansion board operatively coupled to the processor; and  
at least one flash device operatively coupled to the processor, the at least one flash device including firmware instructions which when executed by the processor perform operations comprising:  
initializing the expansion board during a pre-boot phase of the computer system, the expansion board storing a pre-boot driver comprising a pre-boot driver image for the expansion board;  
loading the pre-boot driver image into a memory device of the computer system during the pre-boot phase;  
advertising the pre-boot driver in a data structure of the computer system to indicate the location of the pre-boot driver image in the memory device;  
accessing the hardware device using an OS native driver for the hardware device if the OS native driver is available to the operating system; and  
accessing the hardware device during the OS runtime using the pre-boot driver if the OS native driver for the hardware device is not available to the operating system.

26. (Previously Presented) The computer system of claim 25 wherein the firmware instructions are to operate in accordance with an Extensible Firmware Interface (EFI) framework standard.
27. (Original) The computer system of claim 25 wherein the pre-boot driver image comprises an EFI Byte Code (EBC) image for the expansion board.
28. (Previously Presented) The computer system of claim 25 wherein the data structure is compatible with an Extensible Firmware Interface (EFI) framework standard.